# LINEAR REGRESSION WITH AMAZON AWS MACHINE LEARNING



Here we show how to use Amazon AWS Machine Learning to do linear regression. In a previous post, we explored using <u>Amazon AWS Machine Learning for Logistic Regression</u>.

To review, **linear regression** is used to predict some value y given the values x1, x2, x3, ..., xn. in other words it finds the coefficients b1, b2, b3, ..., bn plus an offset c to yield this formula:

$$y = b1x1 + b2x2 + b3x3 + .... + c.$$

It uses the **least squares error** approach to find this formula. In other words, think of all these values x1, x2, ... existing in some N-dimensional space. The line y is the line that minimizes the distance between the observed and predicted values for all these values. So it is the line that most nearly split right down the middle of the data observed in the training set. Since we know what that line looks like, we can take any new data, plug those into the formula, and then make a prediction.

As always models are built like this:

- Take an input set of data that you think it correlated. Such as hours of exercise and weight reduction.
- Split that data into a training set and testing set. Amazon does that splitting for you.
- Run the linear regression algorithm to find the formula for y. Amazon picks linear regression based upon the characteristics of the data. It would pick another type of regression or classification model is we picked a data set that for which that was a better fit.

- Check how accurate the model is by taking the square root of the differences between the observed and predicted values. Amazon actually uses the mean of this difference.
- Then take new data and apply the formula y to make a prediction.

# **Get Some Data**

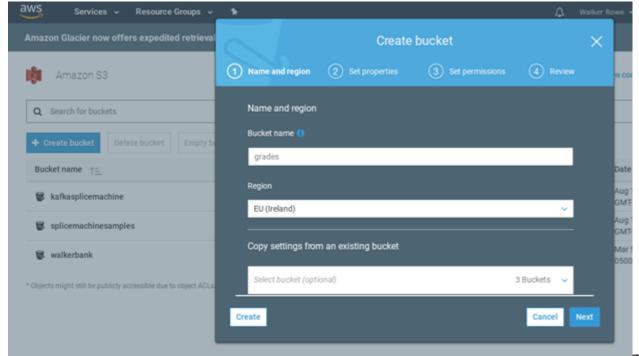
We will use this data of student test scores from the UCI Machine Learning repository.

I copied this data into Google Sheets <u>here</u> so that you can more easily read it. Plus I show the training data set and the one used for prediction.

You download this data in raw format and upload it to Amazon S3. But first, we have to delete the column headings and change the semicolon (;) separators to commas (,) as shown below. We take the first 400 rows as our model training data and the last 249 for prediction. Use vi to delete the first from the data as Amazon will not read the schema automatically (Too bad it does not).

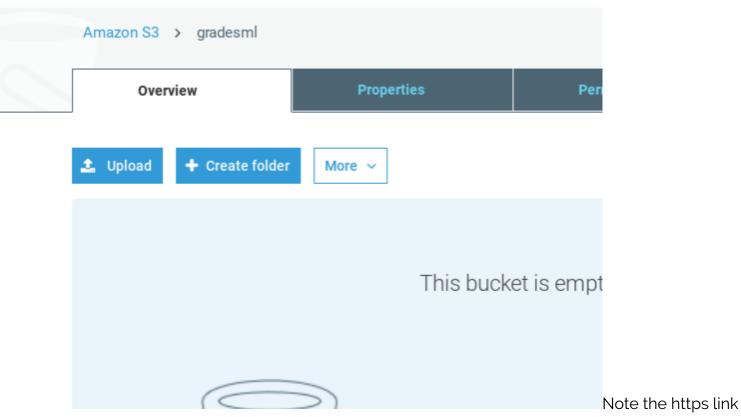
```
vi student-por.csv
sed -i 's/;/,/g' student-por.csv
head -400 student-por.csv > grades400.csv
tail -249 student-por.csv > grades249.csv
```

Now create a bucket in S3. I called it gradesml. Call yours some different name as it appears bucket names have to be unique across all of S3.



Then upload all

3 files.



and make sure the permissions are set to **read**.



#### Owner

critique\_americain

#### Last modified

Mar 15, 2018 11:35:12 AM GMT-0400

#### Etag

973e7bacf3a16bd16e92185dfd64706b

### Storage class

Standard

### Server side encryption

None

#### Size

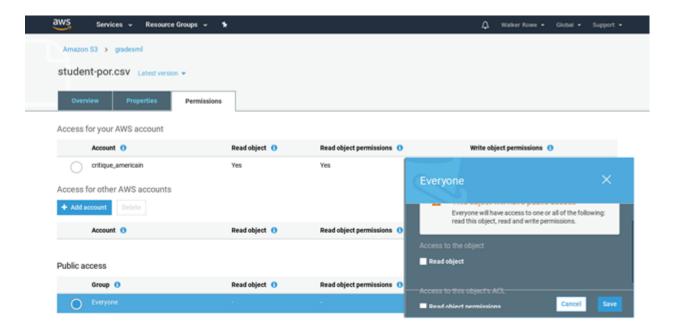
93220

#### Link

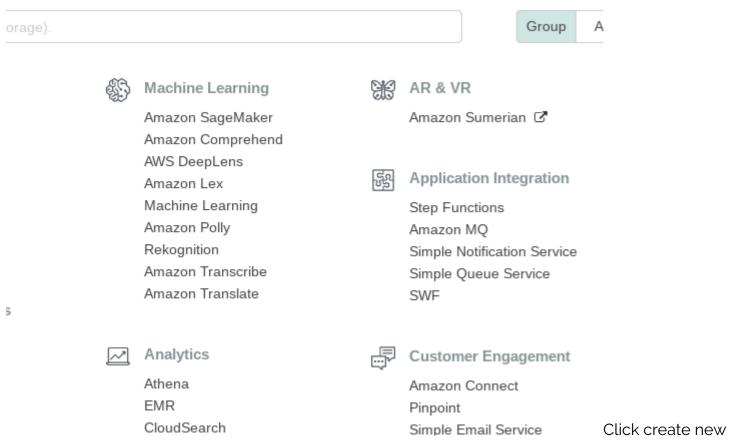
https://s3-eu-west-1.amazonaws.com/gradesml/student-por.csv

Give read

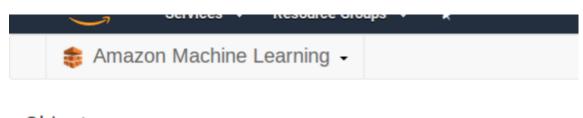
permissions:



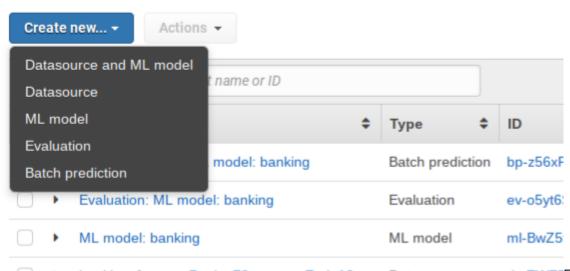
Amazon Machine Learning and then Create New Data Source/ML Model. If you have not used ML before it will ask you to sign up. Creating and evaluating models is free. Amazon charges you for using them to make prediction on a per 1,000 record basis.



Datasource and ML model.

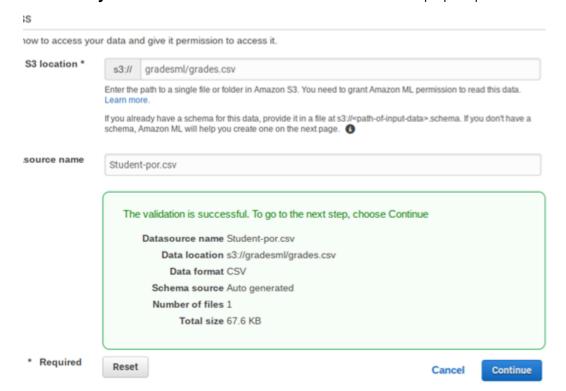


## Objects



banking [percentBegin=70. percentEnd=10... Datasource ds-TWFEFill in the S3 location below. Notice that you do not use the URL. Intead, put the bucket name and file name:

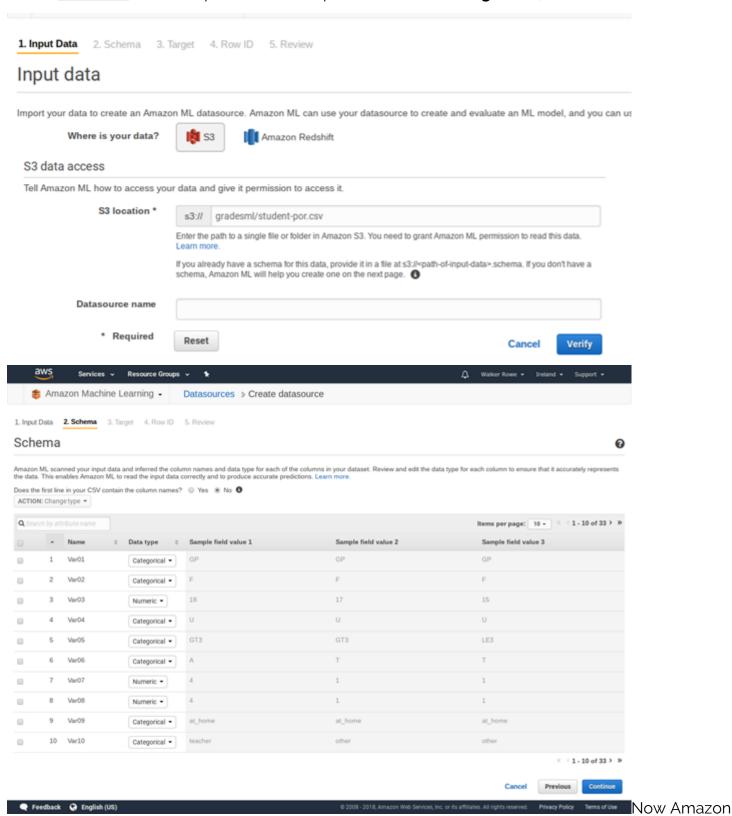
Click verify and Grant Permissions on the screen that pops up next.



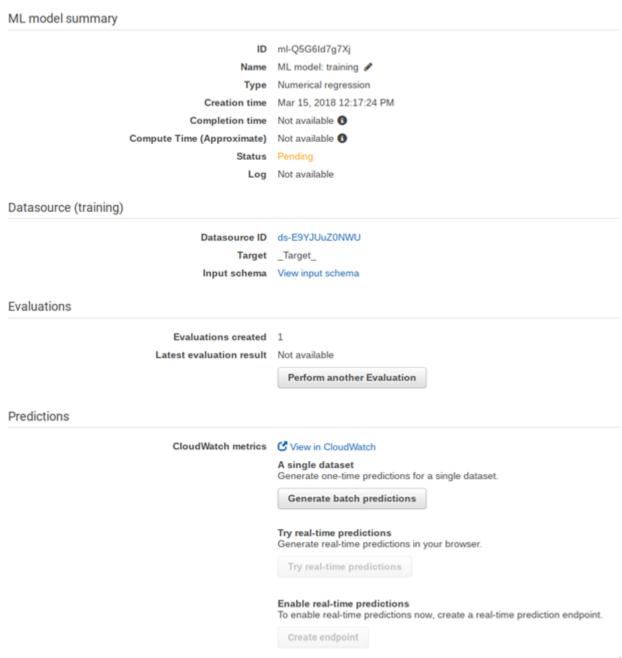
Give the data

source some name then click through the screens. It fill make up field names (we actually don't care what names it uses since we know what each column means from the original data set). It will also determine whether each value is categorical (drawn from a finite set) or just a number. What is important for you to do is to pick the **target**. That is the dependant value you want it to predict, i.e., **y**. From the input data **student-por.csv** pick **G3**, as that is the student's final grade. These grades are from the Portuguese grammar school system and 13 is the highest value.

Below **don't use** students-por.csv as the input data. Instead use **grades400.csv**.

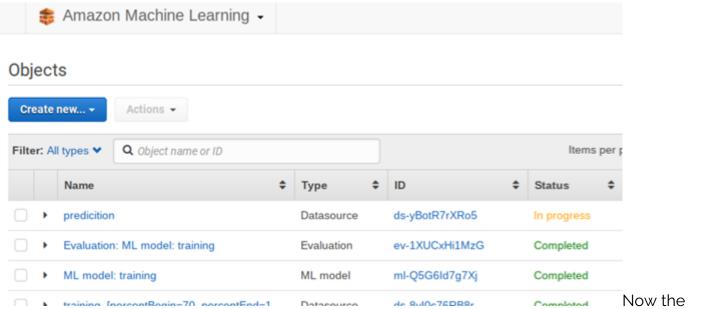


builds the model. This will take a few minutes.

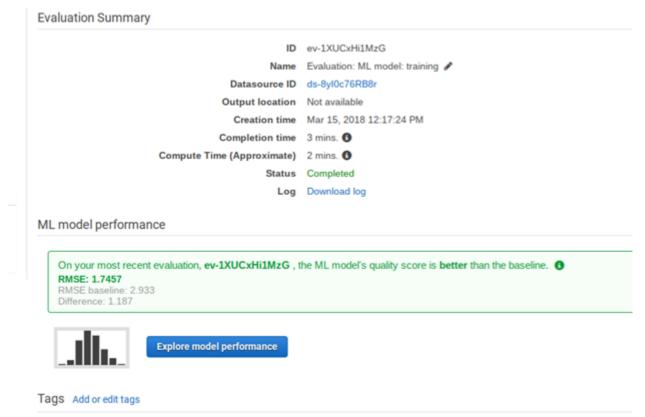


While waiting

are create another **data set**. This is not a model so it will not ask you for a target. Use the **grades249.csv** file in S3, which we will use in the **batch prediction** step.



evaluation is done. We can see which one it is from the list above as it says **evaluation**. Click on it. We explain what it means below.



Amazon shows

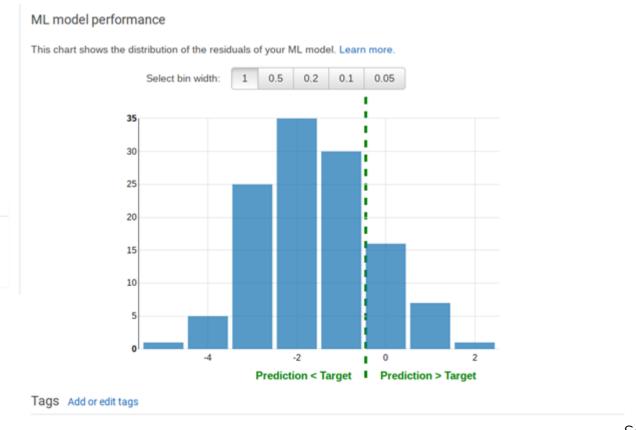
the RMSE. This is the square root of the sum of the squared differences of the observed and predicted values. We square and then take the square root so that all the numbers are positive, so they do not cancel each other out. Amazon also uses the mean, meaning average, by multiplying this sum by 1 / n, where n is the sample size.

If the model and the evaluations were the same, this number would be 0. So the closer to 0 zero we get the more accurate is our model. If the number is large, then the problem is not the algorithm, it is the data. So we could not pick another algorithm to make it much better. There is really only one algorithm used for LR, finding the least squares error. (There are more esoteric ones.) If MSE number is large then either the data is not correlated or, more like, most of the data is correlated, but some

of it is not and is thus messing up our model. What we would do is drop some columns out and rebuild out model to get a more accurate model.

What value means the model is good? The model is good when the distribution of errors is a normal distribution, i.e., the bell curve.

Put another way, click **Explore Model Performance**.



No tans

See the histogram above. Numbers to the left of the dotted line are where the predicted values were

histogram above. Numbers to the left of the dotted line are where the predicted values were less than the observed ones. Numbers to the right are where they are higher. If this distribution were entered on the number 0 then we would have a completely random distribution. That is the idea situation where our errors are distributed randomly. But since it is shifted there is something in our data that we should leave out. For example, family size might not be correlated to grades.

Above Amazon showed the **RMSE baseline**. This is what the RMSE would be if we could have an input data set in which there was this perfect distribution of errors.

Also here we see the limitations of doing this kind of analysis in the cloud. If we have written our own program we could have calculated other statistics that showed exactly which column was messing up our model. Also we could try different algorithms to get rid of the bias caused by **outliers**, meaning numbers far from the mean that distort the final results.

# **Run the Prediction**

Now that the model is saved, we can use it to make predictions. In other words we want to say given these student characteristics what are their likely final grades going to be.

Select the **prediction** datasource you created above then select **Generate Batch Predictions**. Then click through the following screens.

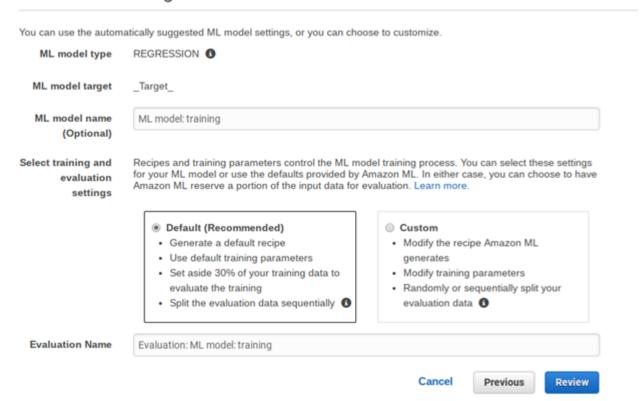
Tags

You selected ML model ml-Q5G6ld7g7Xj. To go to the next step, choose Continue



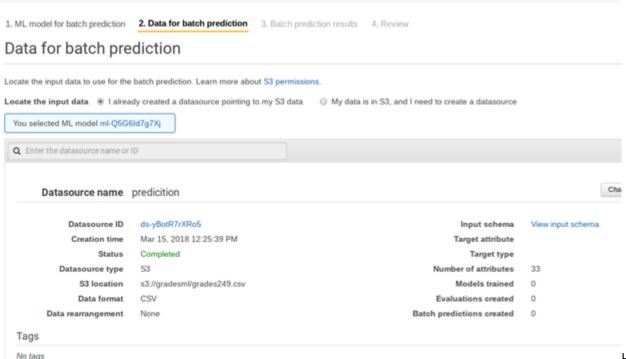
ID ds-yBotR7rXRo5 predicition & Name Creation time Mar 15, 2018 12:25:39 PM 4 mins. Completion time Compute Time (Approximate) 13 mins. **1** Status Completed Message Not available Input schema View input schema Download log Log Use this datasource to ▼ Create (train) an ML model Evaluate an ML model S3 location Generate batch predictions Number of files Data format CSV Total size 34.6 KB Data rearrangement 1. ML model for batch prediction 2. Data for batch prediction 3. Batch prediction results 4. Review ML model for batch prediction Choose the ML model to use for generating batch predictions. Batch predictions generate predictions all at once for a large number of data records Q Search All ML models by name or ID Change ML model ML model name ML model: training ML model ID ml-Q5G6ld7g7Xj Input schema View input schema ML model type Numerical regression Target attribute \_Target\_ Creation time Mar 15, 2018 12:17:24 PM Target type NUMERIC Status Completed Number of attributes 33 Datasource ID ds-E9YJUuZ0NWU Evaluations created 1 Latest evaluation 1.746 (RMSE) result Log Download log Batch predictions 0 created

### ML model settings



Click review

#### then create ML model.



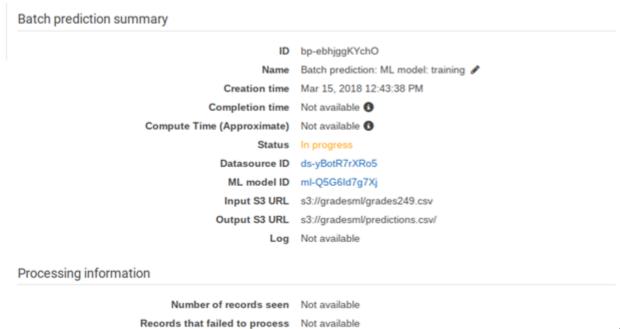
Here we tell it

where to save the results in S3. There it will save several files. The one we are interested in is the one where it calculates the **score**. It should tack it onto the input data to make it easier to read. But it does not. So I have pasted it into <u>this spreadsheet</u> for you on the sheet called prediction and added back the column headings. I also then added a column to show how the MSE mean squared error is calculated.

1. ML model for batch prediction 2. Data for batch prediction 3. Batch prediction results 4. Review

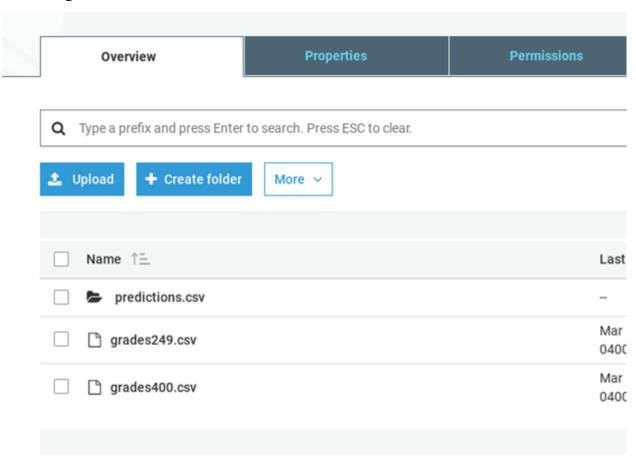
## Batch prediction results

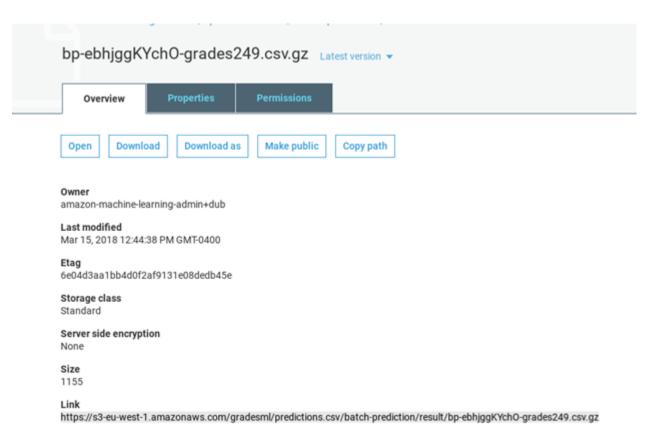
The estimated cost for generating your predictions is \$0.10. This estimate is based on the 249 data records included in your prediction rec The Amazon ML fee for batch predictions is \$0.10 per 1,000 predictions, rounded up to the next 1,000. Learn more. Type the path to the S3 location in which the prediction results will be saved. S3 destination s3:// gradesml/predictions.csv Batch prediction name Batch prediction: ML model: training (Optional) Cancel Previous Review Services → Resource Groups → 1 \$ Amazon Machine Learning - Batch Predictions > Create batch prediction 1. ML model for batch prediction 2. Data for batch prediction 3. Batch prediction results 4. Review Review Review and make any changes, and then click Finish. Edit ML model for batch prediction ML model Name ML model: training ML model ID mi-Q5G6ld7g7Xi Edit Data for batch prediction Data location prediction prediction s3://gradesml/grades249.csv Edit Batch prediction results Output location s3://gradesml/predictions.csv
Batch prediction name Batch prediction: ML model: training Cost Estimate The estimated cost for generating your predictions is \$0.10. This estimate is based on the 249 data records included in your prediction request The Amazon ML fee for batch predictions is \$0.10 per 1,000 predictions, rounded up to the next 1,000. Learn more Tags o Amazon ML copies a maximum of 10 tags from parent objects. Edit the list to keep the tags you need. No tags



As you can see,

it saves the data in S3 in a folder called **predictions.csv**. In this case it gave the prediction values in a file with this long name **bp-ebhjggKYchO-grades249.csv.gz**. You cannot view that online in S3. So download it showing the URL below and look at it with another tool. In this case I pasted the data into Google Sheets.





Download the

#### data like this:

### wget

https://s3-eu-west-1.amazonaws.com/gradesml/predictions.csv/batch-prediction/result/bp-ebhjggKYch0-grades249.csv.gz

Here is that the data looks like with the prediction added to the right to make it easy to see. Column AG is the student's actual grade. AH is the predicted value. Al is the square of the difference. And then at the bottom is MSE.

alth	absences	G1	G2	G3	score	(obs-pred) sqrd
4	4	15	14	17	13.30	13.70998729
5	0	14	13	14	12.18	3.311817626
4	0	11	12	13	12.68	0.1035101929
1	. 10	12	15	15	14.13	0.7519317796
4	4	12	16	16	13.11	8.379114409
5	16	10	11	11	8.30	7.28465498
3	6	10	13	13	11.27	2.991585344
5	0	9	12	12	10.49	2.270657197
3	11	9	11	12	9.44	6.53817229
2	9	13	14	15	12.14	8.19110124
4	0	13	17	17	14.61	5.729368832
4	2	12	15	15	13.06	3.750729422
3	0	14	17	17	15.52	2.18750016
/	21	0	10	10	7 26	7 5001093/6